Spring 2011

CS 740: Complexity Theory and Algorithm Analysis

Pascal Hitzler
Wright State University - Main Campus, pascal.hitzler@wright.edu

Complexity Theory and Algorithm Analysis

CS 740
Wright State University
Spring 2011

Brief Description:
What does it mean to say that some computational problem is intrinsically more difficult than some other problem? How can I claim that I have found a good algorithmic solution? The study of these questions gives rise to an area of Theoretical Computer Science called Complexity Theory, which is based on a systematic and thorough formal study of the complexity of problems with respect to their algorithmic solvability, using Turing machines as main conceptual tool. In this class, we will understand how problem and algorithm complexity is measured, and discuss some of the main complexity classes arising from this study. In particular, we will cover the classes P and NP, and their relationship.

Student Learning Outcomes:
Students acquire an in-depth knowledge of the fundamentals of complexity theory which enables them to understand that some problems are inherently computationally expensive. They also learn how to analyze problems and algorithms with respect to their computational complexities.

Instructor:
Dr. Pascal Hitzler, 389 Joshi.
pascal@pascal-hitzler.de, http://www.knoesis.org/pascal
Office hours: Wednesdays 4:30pm to 5:30pm and by appointment.
Please use email as main means of communication with me outside class.

Class Hours:
Mondays and Wednesdays 6:05pm to 7;20pm, Biological Sciences 105.

Course Materials:
Required: Thomas S. Sudkamp, Languages and Machines, Addison Wesley, 3rd Edition, 2006.
Supplementary: Michael R. Garey and David S. Johnson, Computers and Intractability, Freeman, 1979.

Method of Instruction:
Lecture

Evaluation:
Homework (20%), mid-term exam (30%), final exam (50%)
Grading will follow a standard scale (A: 100-90, B: 89-80, C: 79-70, D: 69-60, F: 59-0). These may be adjusted in favor of the students.

Course Outline:

Week 1	Introduction, Big-Oh-notation
Week 2	Turing machines and complexity
Week 3	Turing machines variations
Week 4	Complexity as a realistic measure
Week 5	Nondeterminism and NP
Week 6	Cook's Theorem
Week 7	NP completeness and reductions
Week 8	Approximation algorithms
Week 9	Beyond NP
Week 10	What we have learned